Login : Jurnal Teknologi Komputer
ISSN : 2302-9692 (print) | 2723-8695 (Online)
Vol. 13, No. 1, June 2019, pp. 26-31
LOGIN http://login.seaninstitute.org/index.php/Login

Comparison of Breadth First Search (BFS) and Depth-First Search (DFS) Methods on File Search in Structure Directory Windows

Sakaria Efrata Ginting[1], Abdul Sani Sembiring[2]
[1]Information Systems Study Program, Sekolah Tinggi Ilmu Komputer Medan, Indonesia,
[2]Informatics Management Study Program, STMIK Budi Darma, Indonesia

Article Info

Article history:

Received, Apr 05, 2019
Revised, Apr 22, 2019
Accepted, May 22, 2019

Keywords:

Directory,
File,
Breadth first search,
Depth first search.

ABSTRACT

Documents of any kind are of course stored by a certain method, with the hope that if in the future the data/documents contained in them are needed then simply searching, the desired data will be obtained quickly. However, the more documents that are stored, the search time will also increase, coupled with incorrect search results. This is due to the large number of documents that must be selected and examined for their relevance to the subject being sought. Breadth First Search (BFS) and Depth-First Search (DFS) is a technique used to search for data in a particular file in a file. With this searching technique, if you want to find one of the data from a lot of data manually, it will take a very long time, but using the Breadth First Search (BFS) and Depth-First Search (DFS) methods, the search will be more fast.

This is an open access article under the CC BY-SA license.

Corresponding Author:

Sakaria Efrata Ginting,
Information Systems Study Program,
Sekolah Tinggi Ilmu Komputer Medan,
Jl. Jamin Ginting No.285, Kwala Bekala, Kec. Medan Johor, Kota Medan, Sumatera Utara 20131.
Email: sakariaginting1983@gmail.com

1. INTRODUCTION

Documents of any kind are of course stored by a certain method, with the hope that if in the future the data/documents contained in them are needed then simply searching, the desired data will be obtained quickly. However, the more documents that are stored, the longer the search time will be, coupled with incorrect search results. This is due to the large number of documents that must be selected and their suitability checked with the subject being sought.

Breadth-First Search (BFS) and Depth-First Search (DFS) is a technique used to find data in a particular file in a file. With this search technique, if you are going to find one of the data from a lot of data manually, it will take a very long time, but by using the Breadth-First Search (BFS) and Depth-First Search (DFS) methods, the search will be faster[1].

Breadth-First Search (BFS) algorithm or wide search is one of the traversal algorithms in a graph. BFS (Breadth-First Search) is an algorithm that performs a wide search that visits a node pre-order, that is, visiting a node then visiting all the nodes that are adjacent to that node first. From an algorithm point of view, all child nodes are obtained by extending the vertices added to the FIFO (First In First Out) queue. Search always visits the tree nodes in a wide manner, starting from a level with a depth of 0 to a maximum depth. This can be expressed in a queue. Each visited node enters the queue only once[2].

The DFS algorithm is also a traversal algorithm in a graph. DFS (Depth First Search) means. is an in-depth search. The depth-first search is a search technique by tracing the deepest point of a tree. This technique visits all nodes in the existing tree first regardless of the weight on each node. After a node in a certain section has been visited and has not achieved the desired destination, backtracking will be carried out to the other node that has not been visited. This technique can be implemented using a tree-based search with last in first out (LIFO) queues on the stack or using recursive functions[3].

2. RESEARCH METHOD

In carrying out this research, clear and structured stages are needed, in order to facilitate the process, it is necessary to make a diagram design such as the diagram below:

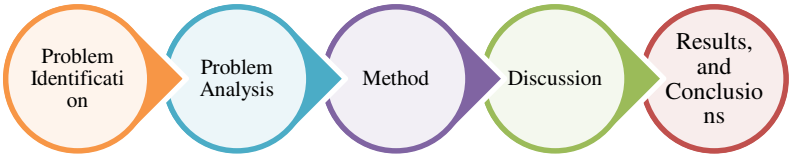

Figure 1. Diagram of Methods and Research Stages

In the stages of the research method, the author conducted interviews with experts to obtain symptoms of worms in livestock.

2.1. Basic theory

A. Breadth First Search (BFS)

BFS (Breadth-First Search) is an algorithm that performs a wide search that visits a node pre-order, that is, visiting a node then visiting all the nodes that are adjacent to that node first. From an algorithm point of view, all child nodes are obtained by extending the vertices added to the FIFO (First In First Out) queue[4].

Search always visits the tree nodes in a wide manner, starting from a level with a depth of 0 to a maximum depth. This can be expressed in a queue. Each visited node enters the queue only once. The algorithm form is as follows[5]:

1. Visit vertex v (if vertex v wants to be visited first).
2. Visit all adjacent vertices of vertex v first.
3. Visit the vertices that have not been visited and are neighboring with the previously visited nodes, and so on.

If the graph is a rooted shape, then all vertices at level d are visited first before visiting vertices at level d + 1. Figure 1 is an example of BFS in the form of a graph search tree.

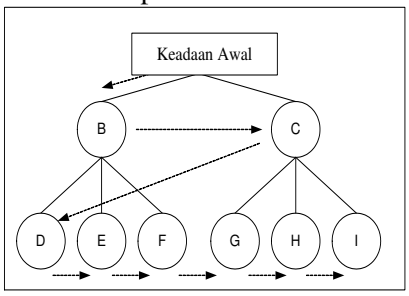

Figure 1.Breadth First Search Procedure

B. Depth First Search (DFS)

The first deep search is a technique of tracing data on vertically and deeply defined nodes. The checking process will move down if the currently inspected node is not suitable for the purpose. The nodes are not extensible sideways even though they still have some extensible child nodes. New child nodes are developed again after the completion of an in-depth search of previously developed child nodes. This process is repeated until a solution is found[6].

The advantage of this method is that it requires a relatively small amount of memory because only the nodes on the active path are stored. Coincidentally, the depth-first search method will find a solution without having to test more in the state space[7].

Whereas the weakness of this method is that it allows not finding the desired goal and will only get 1 solution for each search, as for the First In-depth Search Procedure image can be seen in the image below[8]:

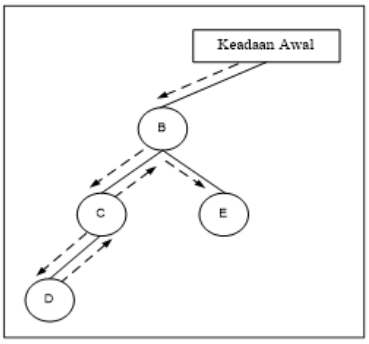

Figure 2.First In-depth Search Procedure

3. RESULTS AND DISCUSSION

The search for file search solutions uses the BFS and DFS search methods, because the solution to the problem can be more than one. Starting from the initial position as the root node, then the BFS and DFS methods seek a solution by developing the root node to the next levels, all possible movements, do not violate the conditions and produce new conditions are developed as much as possible. The search ends when there are no more nodes or new conditions that can be developed. All the nodes which are the destination positions are the solution.

The system analysis stage is carried out after the system planning stage and before the design stage. The analysis stage is a critical and very important stage, because errors in this stage will also cause errors in the next stage, and as an image the directory that is looking for looks like Figure 3 below:

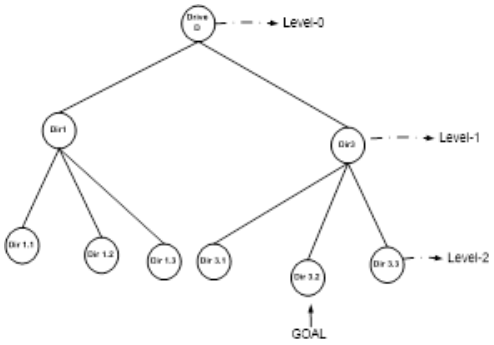

Figure 3.Directory tree

BFS and DFS Algorithm in File Search

To search for files in each directory, it will be done first from the highest directory, namely the "D" drive as seen in Figure 4, while the search process is carried out in the following way:
For example the file to be searched for = "xxxx" contained in the directory "DIR 3.1"

1. Select the highest tree or directory.

Figure 4.Selection of the Highest Directory (root)

2. Select the First Directory as the start of the search process.

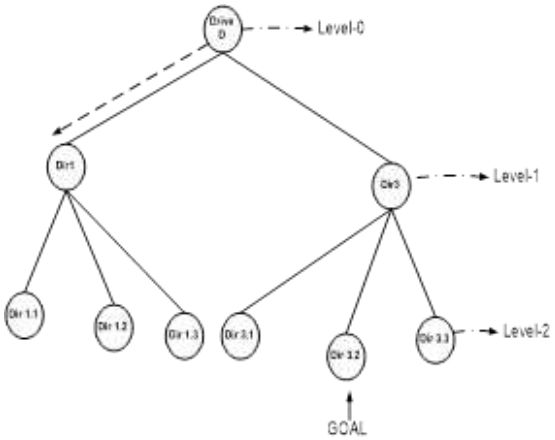

Figure 5. Directory selection at level -1

3. Look in the directory whether there is a file that is "xxxx"? If not, then enter the directory "DIR3".

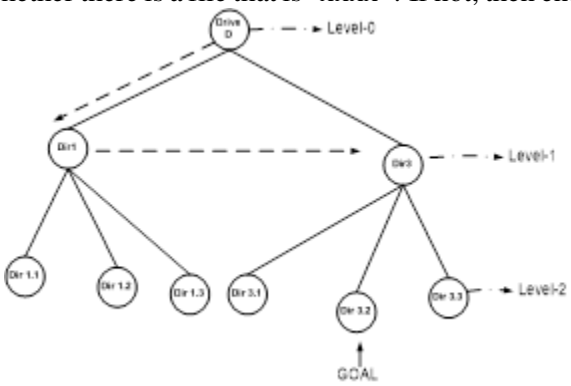

Figure 6.DIR3 Directory Selection

4. Look in the directory "DIR3", is there a file that is "xxxx"? If not, then enter the directory "DIR-1-1 at level 2".

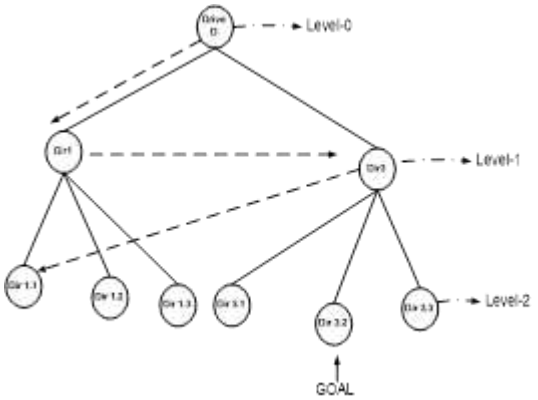

Figure 7.DIR-1-1 Directory Selection

5. Look in the directory "Dir-1-1", is there a file that is "xxxx"? If not, then enter the directory "DIR-1-2".

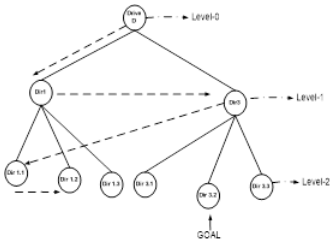

Figure 8.DIR-1-2 Direktory Selection

6. Look in the directory "Dir-1-2", is there a file that is "xxxx"? If not, then enter the directory "DIR-1-3".

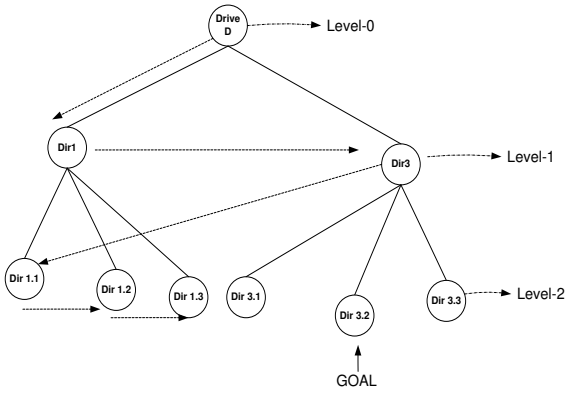

Figure 9.DIR-1-3 Direktory Selection

7. Look in the directory "Dir-1-3", is there a file that is "xxxx"? If not, then enter the directory "DIR-3.1".

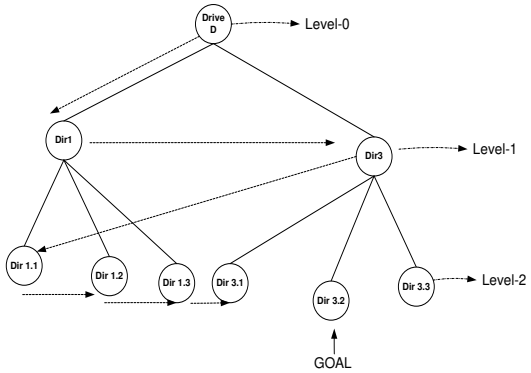

Figure 10.DIR-3-1 Director Selection

8. Look in the directory "Dir-3-1", is there a file that is "xxxx"? If not, then enter the directory "DIR-3.2".

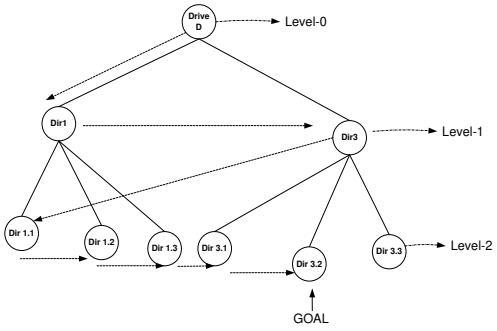

Figure 11.DIR-3-2 Directory Selection

9. Look in the directory "Dir-3-2", is there a file that is "xxxx"? If there is, then display it as GOAL and the search ends.

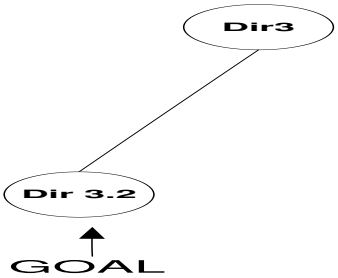

Figure 12.Goal

By looking at the picture above, it is clear that BFS benefits in finding this solution, because BFS visits children widely. So in cases where fewer searches are entered the status tree will stop at that node without going any deeper.

4. CONCLUSION

By implementing the BFS and DFS systems in a document, the process of searching and matching words will be faster and more accurate. In this search process, the best method is BFS with a faster file search.

REFERENCES

[1] B. Prasetiyo and M. R. Hidayah, "Penggunaan Metode Depth First Search (DFS) dan Breadth First Search (BFS) pada Strategi Game Kamen Rider Decade Versi 0.3," *Sci. J. Informatics*, 2015, doi: 10.15294/sji.v1i2.4022.

[2] E. Wijaya, "Analisis Penggunaan Algoritma Breadth First Search Dalam Konsep Artificial Intellegencia," *Time*, 2013.

[3] Bahar and A. Suseno, "Penerapan Metode Depth First Search Pada Sistem Pakar Untuk Mendiagnosa Penyakit Pasca Melahirkan," *Jutisi*, 2014.

[4] R. Apriandi, T. Rismawan, D. M. Midyanti, and J. S. Komputer, "Penerapan Metode Best First Search (Bfs) Untuk Pencarian Lokasi Spbu Terdekat Menggunakan Arduino Berbasis Android," *J. Coding, Sist. Komput. Untan*, 2018.

[5] J. Aldisen, "Aplikasi Penentuan Rute Terpendek Untuk Bagian Pemasaran Produk Roti Surya Dengan Metode BFS," *J. Media Infotama*, 2016.

[6] M. Rahmasuci, H. Hotimatus, M. Azizah, P. Wulandari, D. Adistia, and S. Bukhori, "Strategi Menemukan Jalan Keluar Labirin dengan Waktu Tercepat Menggunakan Metode DFS," 2017.

[7] N. Juliasari and J. C. Sitompul, "Aplikasi Search Engine Dengan Metode Depth First Search (DFS)," *J. Tek. Inform. Univ. Budi Luhur. ISSN 1693 -9166*, 2012, doi: 10.1109/20.312267.

[8] A. Herwanto and B. E. Purnama, "Penerapan Metode Depth First Search Pada Pencarian Rute Bus Kota Berbasis Web Mobile Di Solo," *Ilm. Go Infotech*, 2013.